

B. AMENDMENTS TO THE CLAIMS

1. (Currently Amended) A computer implemented method comprising:
detecting that a memory being managed by a virtual memory manager is constrained; [[and]]
dynamically altering settings used by a sequential access read ahead process in response to the detection, wherein the altered settings are adapted to conserve memory used by the sequential access read ahead process, and wherein the altering includes decreasing a maximum page ahead value, wherein the maximum page ahead value corresponds to a maximum number of pages read by the sequential access read ahead process;
executing the virtual memory manager for a time interval after decreasing the maximum page ahead value;
determining that the memory is less constrained after the time interval has elapsed; and
increasing the maximum page ahead value in response to the determination.
2. (Cancelled)
3. (Currently Amended) The method of claim [[2]] 1 further comprising:
retrieving a value corresponding to a number of free page frames currently being managed by the virtual memory manager; and
calculating a difference between the number of free page frames value and a minimum desired free page constant,

wherein the maximum page ahead value is decreased by an amount based on the calculated difference.

4. (Cancelled)
5. (Original) The method of claim 1 wherein the altering further comprises:
disabling the sequential access read ahead process.
6. (Currently Amended) The method of claim 5 further comprising:
~~executing the virtual memory manager for a time interval after disabling the sequential access read ahead process, determining that the memory is less constrained after the time interval has elapsed, and~~
enabling the sequential access read ahead process in response to the determination.
7. (Currently Amended) A computer-implemented method of managing memory pages, wherein the memory includes a number of used pages and a number of free pages, said method comprising:
retrieving a freelist value corresponding to the current number of free pages;
determining that the freelist value is less than a predetermined minimum value; [[and]]
dynamically altering settings used by a sequential access read ahead process in response to the detection, wherein the altered settings are adapted to decrease the allocation of free pages to the sequential access read ahead process,

and wherein the altering includes decreasing a current maximum page ahead value, wherein the current maximum page ahead value corresponds to a maximum number of pages read by the sequential access read ahead process, and wherein the decreasing further includes:

calculating a shift value based on the difference between the freelist value and the predetermined minimum value; and

bit shifting a constant maximum page ahead value by the shift value, the bit shifting resulting in the current maximum page ahead value.

8. (Cancelled)

9. (Cancelled)

10. (Currently Amended) The method of claim [[9]] 7 wherein the calculating further comprises:

dividing the difference between the freelist value and the predetermined minimum value by the difference between the predetermined minimum value and a predetermined adjustment threshold, the dividing resulting in a quotient and a remainder; and

setting the shift value to the quotient.

11. (Currently Amended) The method of claim [[8]] 7 further comprising:

executing a virtual memory manager for a time interval following the decreasing, wherein the virtual memory manager manages the memory pages and wherein the virtual

memory manager includes the sequential access read ahead process;

retrieving a subsequent freelist value corresponding to the number of free pages available after the time interval;

comparing the subsequent freelist value with the current maximum page ahead value; and

disabling the sequential access read ahead process in response to the comparison.

12. (Currently Amended) The method of claim [[8]] 7 further comprising:

executing a virtual memory manager for a time interval following the decreasing, wherein the virtual memory manager manages the memory pages and wherein the virtual memory manager includes the sequential access read ahead process;

retrieving a subsequent freelist value corresponding to the number of free pages available after the time interval;

determining that the subsequent freelist value is less than the predetermined minimum value;

calculating a second shift value based on the difference between the subsequent freelist value and the predetermined minimum value; and

bit shifting the constant maximum page ahead value by the second shift value, the bit shifting resulting in the current maximum page ahead value.

13. (Currently Amended) The method of claim [[8]] 7 further comprising:
executing a virtual memory manager for a time interval following the decreasing, wherein the virtual memory manager manages the memory pages and wherein the virtual memory manager includes the sequential access read ahead process;
retrieving a subsequent freelist value corresponding to the number of free pages available after the time interval;
determining that the subsequent freelist value is greater than the predetermined minimum value; and
setting the current maximum read ahead value equal to the constant maximum page ahead value in response to the determination.
14. (Original) The method of claim 7 wherein the altering further comprises:
disabling the sequential access read ahead process.
15. (Currently Amended) An information handling system comprising:
one or more processors;
a memory accessible by the processors;
an operating system that controls the processors;
a virtual memory manager, included with the operating system, that manages use of the memory;

a nonvolatile storage area managed by the operating system and including a disk swap area used by the virtual memory manager;

a sequential access read ahead process performed by the operating system adapted to pre-fetch data being sequentially read from file stored in the nonvolatile storage area;

memory conservation software used by the virtual memory manager, the software effective to:

detect that a memory being managed by a virtual memory manager is constrained; [[and]]

dynamically alter settings used by the sequential access read ahead process in response to the detection, wherein the altered settings are adapted to conserve memory used by the sequential access read ahead process, and wherein the software is further effective to decrease a maximum page ahead value, wherein the maximum page ahead value corresponds to a maximum number of pages read by the sequential access read ahead process;

execute the virtual memory manager for a time interval after decreasing the maximum page ahead value;

determine that the memory is less constrained after the time interval has elapsed; and

increase the maximum page ahead value in response to the determination.

16. (Cancelled)

17. (Currently Amended) The information handling system of claim [[16]] 15 wherein the software is further effective to:
retrieve a value corresponding to a number of free page frames currently being managed by the virtual memory manager; and
calculate a difference between the number of free page frames value and a minimum desired free page constant, wherein the maximum page ahead value is decreased by an amount based on the calculated difference.
18. (Cancelled)
19. (Original) The information handling system of claim 15 wherein the software is further effective to:
disable the sequential access read ahead process.
20. (Currently Amended) The information handling system of claim 19 wherein the software is further effective to:
~~execute the virtual memory manager for a time interval after disabling the sequential access read ahead process,~~
~~determine that the memory is less constrained after the time interval has elapsed;~~ and
enable the sequential access read ahead process in response to the determination.
21. (Currently Amended) An information handling system comprising:
one or more processors;
a memory accessible by the processors;

an operating system that controls the processors;

a virtual memory manager, included with the operating system, that manages use of the memory;

a nonvolatile storage area managed by the operating system and including a disk swap area used by the virtual memory manager;

a sequential access read ahead process performed by the operating system adapted to pre-fetch data being sequentially read from file stored in the nonvolatile storage area;

memory conservation software used by the virtual memory manager, the software effective to:

retrieve a freelist value corresponding to the current number of free pages;

determine that the freelist value is less than a predetermined minimum value; [[and]]

dynamically alter settings used by a sequential access read ahead process in response to the detection, wherein the altered settings are adapted to decrease the allocation of free pages to the sequential access read ahead process, and wherein the altering includes decreasing a current maximum page ahead value, wherein the current maximum page ahead value corresponds to a maximum number of pages read by the sequential access read ahead process, wherein the decreasing includes:

calculating a shift value based on the difference between the freelist value and the predetermined minimum value; and

bit shifting a constant maximum page ahead value by the shift value, the bit shifting resulting in the current maximum page ahead value.

22. (Cancelled)
23. (Cancelled)
24. (Currently Amended) The information handling system of claim [[23]] 21 wherein the calculating further comprises: dividing the difference between the freelist value and the predetermined minimum value by the difference between the predetermined minimum value and a predetermined adjustment threshold, the dividing resulting in a quotient and a remainder; and
setting the shift value to the quotient.
25. (Currently Amended) The information handling system of claim [[22]] 21 further comprising:
executing a virtual memory manager for a time interval following the decreasing, wherein the virtual memory manager manages the memory pages and wherein the virtual memory manager includes the sequential access read ahead process;
retrieving a subsequent freelist value corresponding to the number of free pages available after the time interval;
comparing the subsequent freelist value with the current maximum page ahead value; and
disabling the sequential access read ahead process in response to the comparison.

26. (Currently Amended) The information handling system of claim [[22]] 21 further comprising:
executing a virtual memory manager for a time interval following the decreasing, wherein the virtual memory manager manages the memory pages and wherein the virtual memory manager includes the sequential access read ahead process;
retrieving a subsequent freelist value corresponding to the number of free pages available after the time interval;
determining that the subsequent freelist value is less than the predetermined minimum value;
calculating a second shift value based on the difference between the subsequent freelist value and the predetermined minimum value; and
bit shifting the constant maximum page ahead value by the second shift value, the bit shifting resulting in the current maximum page ahead value.

27. (Currently Amended) The information handling system of claim [[22]] 21 further comprising:
executing a virtual memory manager for a time interval following the decreasing, wherein the virtual memory manager manages the memory pages and wherein the virtual memory manager includes the sequential access read ahead process;
retrieving a subsequent freelist value corresponding to the number of free pages available after the time interval;
determining that the subsequent freelist value is greater than the predetermined minimum value; and

setting the current maximum read ahead value equal to the constant maximum page ahead value in response to the determination.

28. (Original) The method of claim 27 wherein the altering further comprises:
disabling the sequential access read ahead process.
29. (Currently Amended) A computer program product stored on a computer operable media, said computer program product comprising:
means for detecting that a memory being managed by a virtual memory manager is constrained; [[and]]
means for dynamically altering settings used by a sequential access read ahead process in response to the detection, wherein the altered settings are adapted to conserve memory used by the sequential access read ahead process, and wherein the means for altering includes means for decreasing a maximum page ahead value, wherein the maximum page ahead value corresponds to a maximum number of pages read by the sequential access read ahead process;
means for executing the virtual memory manager for a time interval after decreasing the maximum page ahead value;
means for determining that the memory is less constrained after the time interval has elapsed; and
means for increasing the maximum page ahead value in response to the determination.
30. (Cancelled)

31. (Currently Amended) The computer program product of claim [[30]] 29 further comprising:
means for retrieving a value corresponding to a number of free page frames currently being managed by the virtual memory manager; and
means for calculating a difference between the number of free page frames value and a minimum desired free page constant, wherein the maximum page ahead value is decreased by an amount based on the calculated difference.
32. (Cancelled)
33. (Original) The computer program product of claim 29 wherein the means for altering further comprises:
means for disabling the sequential access read ahead process.
34. (Currently Amended) The computer program product of claim 33 further comprising:
~~means for executing the virtual memory manager for a time interval after disabling the sequential access read ahead process;~~
~~means for determining that the memory is less constrained after the time interval has elapsed; and~~
means for enabling the sequential access read ahead process in response to the determination.
35. (Currently Amended) A computer-implemented computer program product for managing memory pages, wherein the memory

includes a number of used pages and a number of free pages, said computer program product comprising:
means for retrieving a freelist value corresponding to the current number of free pages;
means for determining that the freelist value is less than a predetermined minimum value; [[and]]
means for dynamically altering settings used by a sequential access read ahead process in response to the detection, wherein the altered settings are adapted to decrease the allocation of free pages to the sequential access read ahead process, and wherein the means for altering includes means for decreasing a current maximum page ahead value, wherein the current maximum page ahead value corresponds to a maximum number of pages read by the sequential access read ahead process, and wherein the means for decreasing includes:

means for calculating a shift value based on the difference between the freelist value and the predetermined minimum value; and

means for bit shifting a constant maximum page ahead value by the shift value, the bit shifting resulting in the current maximum page ahead value.

36. (Cancelled)
37. (Cancelled)
38. (Currently Amended) The computer program product of claim [[37]] 35 wherein the means for calculating further comprises:

means for dividing the difference between the freelist value and the predetermined minimum value by the difference between the predetermined minimum value and a predetermined adjustment threshold, the dividing resulting in a quotient and a remainder; and
means for setting the shift value to the quotient.

39. (Currently Amended) The computer program product of claim [[36]] 35 further comprising:
means for executing a virtual memory manager for a time interval following the execution of the means for decreasing, wherein the virtual memory manager manages the memory pages and wherein the virtual memory manager includes the sequential access read ahead process;
means for retrieving a subsequent freelist value corresponding to the number of free pages available after the time interval;
means for comparing the subsequent freelist value with the current maximum page ahead value; and
means for disabling the sequential access read ahead process in response to the comparison.
40. (Currently Amended) The computer program product of claim [[36]] 35 further comprising:
means for executing a virtual memory manager for a time interval following the execution of the means for decreasing, wherein the virtual memory manager manages the memory pages and wherein the virtual memory manager includes the sequential access read ahead process;

means for retrieving a subsequent freelist value corresponding to the number of free pages available after the time interval;

means for determining that the subsequent freelist value is less than the predetermined minimum value;

means for calculating a second shift value based on the difference between the subsequent freelist value and the predetermined minimum value; and

means for bit shifting the constant maximum page ahead value by the second shift value, the bit shifting resulting in the current maximum page ahead value.

41. (Currently Amended) The computer program product of claim [[36]] 35 further comprising:
means for executing a virtual memory manager for a time interval following the decreasing, wherein the virtual memory manager manages the memory pages and wherein the virtual memory manager includes the sequential access read ahead process;
means for retrieving a subsequent freelist value corresponding to the number of free pages available after the time interval;
means for determining that the subsequent freelist value is greater than the predetermined minimum value; and
means for setting the current maximum read ahead value equal to the constant maximum page ahead value in response to the determination.

42. (Original) The computer program product of claim 35 wherein
the means for altering further comprises:
means for disabling the sequential access read ahead
process.